

Challenges and Development Trends in Cloud-Native Applications: A Comprehensive Survey

Sandeep K Guduru

Abstract—The core value that cloud-native applications (CNAs) deliver through modern software development depends upon cloud computing's natural scalability, adaptability, and automated systems. The paper dives deeply into CNAs while analyzing their foundational concepts that combine service-based designs, API management, and infrastructure code automation for expedited implementation and resilient dynamic operations. The growth of CNAs tracks from original cloud computing models through modern artificial intelligence systems that include edge computing and multi-cloud implementations. An examination of new trends in microservices serverless computing and containerization and DevSecOps details how they contribute to system scalability and operational efficiency while improving security. Cloud-native applications continue to evolve, but professionals encounter complexity alongside interoperability issues and implementation lock-ins that need innovative approaches. The research investigates sustainability effects alongside cost optimization practices in cloud-native environments to offer practical guidance for companies working on their cloud-native architecture improvement. The analysis provides comprehensive insights into Cloud Native Applications to empower developers and decision-makers in achieving cloud computing success through understanding complex trends and current challenges.

Keywords—Cloud-native Applications, Cloud Computing, Orchestration Platforms, Development Trends, Application Deployment.

I. INTRODUCTION

An approach to application development and operation that is "cloud-native" makes full use of the benefits offered by the Cloud Computing (CC) delivery paradigm [1]. New guidelines for the creation and implementation of apps are provided by cloud-native [2]. A software as a service application may be described using twelve-factor app patterns. The patterns emphasize rapidity, security, and scalability. Being cloud-native often means reducing the technical risk of an application's deployment while simultaneously improving its speed, scalability, and resilience. Another definition of Cloud native is agnostic. Not dependent on any particular system, operating system, programming language, etc. Vendor lock-in is reduced by adopting this mentality. As far as the developer is concerned, all resources are just cloud-based. An open organization called the Cloud-native Computing Foundation (CNCF) has laid up a road map that gives an outline of the steps to take in order to adopt a Cloud-native architecture[3]. A crucial and required component is the orchestration of specific IT jobs executed in containers. With more and more containers being used, orchestration becomes more important. Another feature of container orchestrators is the ability to remotely control containers. During runtime, orchestrators may affect the lifecycle of a Cloud-native application and aid users in building, scaling, and managing complex applications.

Some common definitions of observability include (i) metrics gathered by activity monitoring, (ii) logging, and (iii) tracing. Every execution environment that runs natively on the cloud must have observability[4]. All of its operations are

together referred to as observability engineering due to its significance in the software lifecycle. Observability engineering is centered toward deciphering intricate and multi-layered structures. An improved understanding of the Cloud-native system on several levels is the primary goal of observability. The advantages of observability features highlight the skills that may be acquired, such as the ability to detect and analyze abnormal behavior, predict when things will break, get alerts when certain events occur, optimize performance, proactively enhance the customer experience, and much more.

There is an urgent need for a comprehensive study that evaluates the engineering approaches to observability, its maturity, efficiency, tools, and future research paths, even if recent papers suggest that cloud-native observability is a key subject[5]. This work's value lies in the Systematic Mapping Study (SMS) it conducted on the observability of cloud-native applications (CNApps) [6]. SMS addresses every single research question. This leads to a map that depicts the current cutting edge. Integrating observability into cloud-native settings is a choice. Primary studies of sufficient quality, particularly those dealing with logging and tracing, are in short supply in this field of study, likely as a result of this alternative[7]. Advised sending text messages to places with poor service.

A. Motivation and Significance of the Paper

The motivation behind this paper is to explore the evolving landscape of cloud-native applications, which have become integral to modern software development due to their scalability, flexibility, and agility. A growing number of businesses adopting cloud environments face increasing application management complexity that creates important obstacles regarding security and performance monitoring as well as orchestration. The purpose of this article is to provide an overview of the problems that cloud-native apps are now facing, as well as to discuss recent trends in development and new approaches to fixing these problems. In doing so, it hopes to provide helpful information on how to optimize cloud-native architectures, which will allow businesses to make use of cloud computing to its fullest while also fixing the new problems that are affecting its reliability and safety. Below is the significance of the paper are as follows:

- This research discusses CNAs through an in-depth analysis of service-based design components alongside API network solutions and automated infrastructure management.
- Leaders recognize main trends that include microservices as well as AI-infused architectures, serverless computing, and multi-cloud strategies defining cloud-native ecosystems.
- This paper examines how machine learning and AI boost automation and scalability while optimizing performance within cloud-native applications.
- The research document demonstrates how multi-cloud and hybrid cloud solutions bring adaptability alongside dependable infrastructure along with protected computing environments to CNAs.

- The paper addresses challenges like complex architectures, scalability, security, and operational issues faced in cloud-native environments.

B. Structure of the paper

Here is the outline of the paper: A description and summary of cloud-native apps (CNAs) are given in Section II. Section III discusses emerging development trends in CNAs, including architectural patterns and multi-cloud strategies. Section IV addresses the challenges in developing and managing CNAs, such as scalability and security issues. A literature review outlining significant contributions to the area is included in Section V. Results and recommendations for further study make up Section VI.

II. OVERVIEW OF CLOUD-NATIVE APPLICATIONS

Figure 1 below shows the interconnected and multi-faceted aspects of cloud-native apps. In a nutshell, cloud-native approaches (DevOps, software design patterns) make it possible to build and compose the architectural components (microservices, containers, container orchestration) that adhere to the softwareization, automation platforms, migration, and interoperability common principles of CNAs. Lastly, applications that adhere to a cloud-native design naturally possess CNA qualities, such as scalability and elasticity [8].

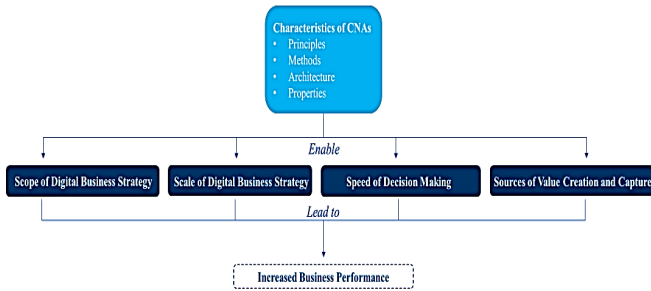


Fig. 1. Conceptual model for strategic implications of CNAs

A. Definition and Core Characteristics:

Technological advancements in cloud computing infrastructures have led to a sharp rise in cloud-native applications [9]. In the past, applications used elastic infrastructure to handle a plethora of services running in containers as microservices; now, cloud-native apps do the same. Cloud infrastructure administration has never been easy. Furthermore, orchestration has become a crucial component of these cloud-native apps as it enables a variety of functions, including scheduling, scalability, anomaly detection, resource management, and more. The following are the main characteristics of cloud-native applications:

1) Service-based Architectures

The building blocks of cloud-native apps are collections of independent (micro)services. Independent creation and operation of each service is made possible since each service in an application exists in its own right. Concurrently services often communicate with one another and with other services inside an application; these services are found by taking use of capabilities offered by the application runtime [17][18]. This opens the door to building cloud-native apps and composing services.

2) API-based Interactions

API-based service-to-service connections are used in cloud-native applications. Each service in an application

exposes its capabilities via an API, and each service, in turn, connects to and uses the APIs of the other services in the application. The APIs used in cloud-native applications should all adhere to widely recognized standards, such as REST over HTTP, and each component should have its own API.8.

3) Infrastructure as Code

Every aspect of cloud-native apps, including deployment, administration, scaling, and monitoring, is highly automated. Infrastructure as code, or machine-readable files that enable the specification of the intended configuration for an application and its components, is usually used to accomplish such automation.

B. Evolution of Cloud-Native Development

1) Early Cloud Computing (2000s):

Cloud computing started from Amazon Web Services (AWS), Microsoft Azure, and Google Cloud with IaaS, and with PaaS[10]. These allowed organizations to transition from traditional physically located data centers to scalable and cost-effective virtualized infrastructure.

2) Rise of Containerization (2013–2015):

Containers created by Docker brought a sea change in the app development as applications could be made portable and deployable on any environment[11]. It also witnessed the change from monoliths to microservices, where the teams started using CI/CD pipelines to get the releases sooner.

3) Kubernetes and Orchestration (2015–2018):

Kubernetes, developed by Google, was able to become the platform for managing containerized applications at scale [12]. It eased the distribution of microservices and the management of resources, while with aids like Istio and Linkerd the notion of the service mesh was initiated to provide organise the conversation, defend and observe microservices.

4) Serverless and Function-as-a-Service (2018–2020):

In Serverless computing came into existence whereby developers only concentrate on the code rather than the infrastructure [13]. AWS Lambda, Azure Functions, and Google Cloud Functions gave event-driven programming effortless and massiveness.

5) Modern Cloud-Native Ecosystems (2020–Present):

Modern cloud-native development now comes with further tools such as AI/ML, edge, and multi-cloud. Tools like Kubernetes-native frameworks and adopting DevSecOps practices help achieve secure, scalable and faster continuum of software applications as environments become complex.

III. DEVELOPMENT TRENDS IN CLOUD-NATIVE APPLICATIONS

Cloud-native applications are now shifting as there are several trends influencing their formation. Microservices architecture is used for scalability and flexibility, and serverless computing is an easy way to avoid thinking about infrastructure [14]. Kubernetes remains the king of the orchestration software, and its supporting tools for the service mesh, observability, and automation expand the list. There are emergent trends related to multi-cloud and hybrid cloud to distribute the workload and avoid dependence on particular cloud provider's solutions; edge computing to integrate with cloud-native systems for low latency and support of IoT applications. DevSecOps enhance security throughout the SDLC phase, whereas AI/ML strength catalyzes new thinking in automation and analytics. Observability suites offer all

stakeholders detailed information on complex distributed applications, while low code no-code help firms move applications forward quickly. Sustainability has also become an important consideration, for example, to increase efficiency and to minimize the effects on the environment in cloud-native architecture.

A. Emerging Architectural Patterns:

The new architectural trends paving the way to the future of software development tackle difficult issues and improve system functionality. Here are some notable patterns:

1) Microservices Evolution: Beyond Decentralization

A key feature of microservices architecture is the separation of application logic into smaller, more deployable services, with each service handling a distinct business function [15]. The services are able to connect with one another using well-defined protocols and APIs, which gives developers the freedom to work on specific components independently. Faster development cycles and better fault separation are two benefits of using microservices, which also increase modularity, adaptability, and scalability.

2) AI-Infused Architectures: Cognitive Design

Cognitive design, which is the inclusion of artificial intelligence into software structures is getting incorporated. AI leads the proactive design of systems by forming or redesigning architectures, fine-tuning or redesigning existing performance, improving systems' security, and empowering decision-making processing.

3) Quantum Computing Integration

Quantum computing harnesses quantum physics to achieve calculations – a computational evolution [16]. These are qubits that are able to be superposed and entangled to alter their states at the flick of a switch across various distances. In this encryption, optimization and materials research, quantum computers may solve complicated operations more efficiently. Multi-cloud solutions therefore involve using lots of cloud providers to provide for flexibility, reliability, and cheapness.

4) Workload distribution across clouds

It helps companies to reduce vendor lock-in, enhances results with the increase in the number of clouds, with the positive impacts on disaster recovery.

5) Serverless Architecture

Serverless means that a developer does not have to manage or even think about servers since cloud providers will handle this on their own. A serverless function runs strictly on the basis of events with no manual tuning of execution, time or resources involved hence it has better scaling, low operational overheads, and costs implications.

6) Hexagonal Architecture (Ports and Adapters)

Software projects may benefit from using hexagonal architecture, which is a method for managing dependency inversion. Other names for this design include ports and adapters and onion architecture [17]. When designing software, hexagonal architecture encourages a laser-like concentration on the domain's primary business logic and gives external integration points a secondary status. You can better manage complicated domains over the long run using hexagonal design, which encourages architectural growth and the adoption of excellent practices like test-driven development (TDD) by software developers.

B. Integration of AI/ML in Cloud-Native Applications

Application development that is "cloud-native" means it was built from the ground up to make the most of cloud computing [18]. The combination of microservices with containerization and dynamic orchestration creates applications which scale efficiently and promptly address changes in customer demand. AI technologies drive a fundamental transformation of cloud-native application development practices. The benefits AI provides developers include automated operations while improving both software quality and offering better scalability for next-generation cloud-native software development. A future of cloud-native application development hinges on an integration of AI. ML, NLP, and other AI techniques are poised to drive innovation, allowing developers to create smarter applications that can learn from data, predict user behavior, and continuously adapt. By leveraging AI, cloud-native development teams can optimize resources, enhance system performance, and build more personalized experiences for users.

C. Multi-Cloud and Hybrid Cloud Strategies

Multi-cloud means the utilization of multiple different public cloud vendors and services that allow organizations access a wide variety of the available resources and solutions. In contrast, the hybrid model uses at least one private cloud solution and one or more public clouds and seeks a balance between security control and cloud flexibility.

Scale Grid plays a critically important role in the contemporary landscape of cloud services because it is BS that is designed to effectively tackle both the multi-cloud environments and also the hybrid cloud infrastructures made of private clouds that can be augmented by several public ones.

Multi-cloud means the utilization of services of different providers to achieve more versatility with fewer vendor-related restrictions, whereas the terms 'hybrid cloud' imply the use of both private and public cloud infrastructure types to achieve more control and manageability while receiving the ability to scale a project.

IV. CHALLENGES IN CLOUD-NATIVE APPLICATIONS

There are some common challenges in CNAs as follows [19][20]:

A. Complex Architecture

Applications designed for the cloud are often microservices-based, which creates a number of challenges for managing a large number of discrete and relatively independent services. Implementing good communication between these services, managing dependencies on other services and synchronizing the execution of these components is not a trivial task especially when the environment is dynamic as is in the cloud.

B. Scalability & Performance

This is a result of the concept of scalability, which is a fundamental characteristic of cloud-native systems, though it is difficult to come by in an efficient manner. Load balancing management, state management in stateless architecture, and managing the network latency between the various components are major challenges.

C. Security

One of the issues that must be addressed in applications designed for the cloud-native environment is security. Security must be extended at different layers thus other important steps include enforcing proper authentication and authorization controls, proper protection of data that is at rest, and the multiple attacks due to multiple microservices and API calls.

1. Operational Issues

The toughest issues related to the actual running of cloud-native applications are utilizing CI/ CD to set up a proper application pipeline, managing the logging and monitoring of a great number of distributed services, and debugging a problem that is present in these highly interconnected services. Operational thus demands advanced instruments for proficient management.

V. LITERATURE REVIEW

This survey focuses on literature review based on cloud-native applications, their characteristics, challenges, and applicability for creating scalable, resilient, and efficient systems.

In this study, Szalay, Mátray and Toka (2021) advocate for a fully functional state layer to bolster the architecture of stateless cloud applications. To reduce the amount of communication between hosts caused by state externalization, one approach is to use a data placement algorithm in conjunction with a system architecture to distribute the states of functions among the hosts in a data center. However, in order to guarantee a sweet spot with low network traffic and short state-access time, they construct a dynamic replication module that determines the appropriate number of copies for each state. They test the suggested approaches in real-world situations[21].

In this study, Jiao, Xu and Fan (2021) create and investigates a Kubernetes-based CNAs architecture to address microservice security and performance monitoring issues. The following features are present in the design: it separates the main application from environment-related tasks like database deployment, performance detection, and service request

security verification, allowing micro-service applications to focus on implementing business logic; Simultaneously, the development cost may be decreased, and the aforementioned functional modules can be reused extensively using the second level container deployment[22].

In this study, Mendki (2021) demonstrates how cloud-native apps may benefit from blockchain technology to tackle security issues. In this study, they examine the problems with and potential solutions to those problems using blockchain technology in the following areas: audit log for forensics, network security, authentication, identity management, and container security. The last ten years have seen cloud computing take center stage. As cloud-based computing has grown in popularity, new architectural styles have emerged. The emergence of cloud-native apps has guaranteed scalability and agility for applications. Regarding safety, nevertheless, they present unique obstacles[23].

In this study, Chowdhury et al. (2020) offer a system that uses machine learning to automate orchestration in advance, taking into account application-related parameters in addition to absolute and relative metrics, in order to facilitate scalability. Applications in the cloud-native era require two fundamental components to manage their life cycle through monitoring and automated orchestration. There are a plethora of monitoring tools out there that can keep tabs on these deployments, but they don't collect data specific to applications, and automated orchestration is in its infancy[24].

In this study, Pradeep Pai and Kumar (2021) explores the characteristics of applications with many components, basic methods for building applications using container-based technologies, and their performance in comparison to deployments based on virtual machines. The paper compares and contrasts two deployment methods, one based on virtual machines and the other on containers, and assesses how well each one handles multi-component applications. Cloud-native applications represent software solutions developed through a primary focus on cloud computing principles. These applications use cloud computing models provided by cloud providers, whether they are on-premise or off-premise, to get better performance, more customization options, and fewer deployment risks[25].

TABLE I. THIS TABLE SUMMARIZES THE LITERATURE REVIEW BASED ON CLOUD-NATIVE APPLICATIONS, FOCUSING ON THEIR AREAS OF RESEARCH, FINDINGS, CHALLENGES, AND KEY CONTRIBUTIONS.

Reference	Focus Area	Key Findings	Challenges	Key Contribution
Szalay, Mátray, and Toka (2021)	Cloud Application Design and State Management	Proposes a state layer design for stateless cloud applications, with a data placement algorithm to minimize inter-host communication.	Balancing state-access time and network traffic while ensuring efficient replication.	Design of a dynamic replication module and a data placement algorithm to optimize state management and communication in cloud applications.
Jiao, Xu, and Fan (2021)	Cloud-Native Application Architecture (Kubernetes)	Designs a Kubernetes-based cloud-native architecture to address performance monitoring and security in microservices.	Integration of environment-related functions and ensuring reusability of modules.	Proposes an architecture that decouples environment-related functions and allows for high reusability, reducing development costs and improving security.
Mendki (2021)	Blockchain for Cloud-Native Security	Specifically looks at blockchain's potential to improve cloud-native app security in the areas of identity management, container security, and network security.	Security challenges in cloud-native environments such as network security, authentication, and audit logs.	Illustrates blockchain-based solutions for securing cloud-native applications, particularly in areas like container security, identity management, and network security.
Chowdhury et al. (2020)	Cloud-Native Application Monitoring and Orchestration	Proposes a framework for automated orchestration using machine learning based on application-related metrics to improve scalability.	Lack of tools that monitor application-related metrics and premature stage of automated orchestration.	Develops a machine learning-based framework for proactive orchestration and monitoring of cloud-native applications to enhance scalability and lifecycle management.

Pradeep Pai and Kumar (2021)	Container vs Virtual Machine-based Deployments	Compares container-based and virtual machine-based deployments for multi-component applications in cloud-native environments.	Performance evaluation of multi-component applications in different deployment environments.	Provides a comparative study of container-based and virtual machine-based deployments, highlighting the advantages of cloud-native application development.
------------------------------	--	---	--	---

VI. CONCLUSION AND FUTURE WORK

Cloud-native applications (CNAs) have revolutionized modern software development with their capabilities to deliver scalability along with flexibility and fast deployment. The analysis investigated the heritage and fundamental elements alongside upcoming trends of CNAs as it examined how microservices and containerization, together with orchestration and serverless computing, operate. The importance of observability, together with multi-cloud strategies and AI/ML integration, emerged as vital elements for managing scalability and security as well as complexity within modern cloud environments. Through their evolution process CNAs position themselves to shape industrial innovation while providing businesses remarkable prospects for operation optimization and improved customer relationships. The increasing intricacy of cloud-native ecosystems requires comprehensive strategies to properly address architectural issues alongside operational limitations.

Future studies should concentrate on developing advanced frameworks and technological systems which improve the observability and security of CNAs while enabling greater automation features. A study of how quantum computing combined with edge computing operates within CNAs holds potential to develop low-latency high-performance applications. Sustainable cloud-native architecture research must advance to decrease environmental impact while increasing efficient operation. Studies concerning multi-cloud and hybrid cloud strategies need to focus on breaking vendor lock-in and developing enhanced disaster recovery capabilities. Analysis of AI and machine learning tools used by CNAs requires more investigation to maximize their predictive capabilities and anomaly detection effectiveness and intelligent decision-making potential.

REFERENCES

- [1] M. Stine, *Migrating to Cloud-Native Application Architectures*. 2014.
- [2] S. Brunner, M. Blochlinger, G. Toffetti, J. Spillner, and T. M. Bohnert, "Experimental Evaluation of the Cloud-Native Application Design," *Proc. - 2015 IEEE/ACM 8th Int. Conf. Util. Cloud Comput. UCC 2015*, pp. 488–493, 2015, doi: 10.1109/UCC.2015.87.
- [3] J. Humble and D. Farley, *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*. 2010.
- [4] J. Kosińska and K. Zieliński, "Autonomic Management Framework for Cloud-Native Applications," *J. Grid Comput.*, 2020, doi: 10.1007/s10723-020-09532-0.
- [5] N. Kratzke, "Cloud-Native Observability: The Many-Faceted Benefits of Structured and Unified Logging—A Multi-Case Study," *Futur. Internet*, 2022, doi: 10.3390/fi14100274.
- [6] K. Petersen, S. Vakkalanka, and L. Kuzniarz, "Guidelines for conducting systematic mapping studies in software engineering: An update," *Inf. Softw. Technol.*, 2015, doi: 10.1016/j.infsof.2015.03.007.
- [7] B. Kitchenham, "Guidelines for performing systematic literature reviews in software engineering," *Tech. report, Ver. 2.3 EBSE Tech. Report. EBSE*, 2007.
- [8] S. Riaz, A. H. Khan, M. Haroon, S. Latif, and S. Bhatti, "Big data security and privacy: Current challenges and future research perspective in cloud environment," in *Proceedings of 2020 International Conference on Information Management and Technology, ICIMTech 2020*, 2020, doi: 10.1109/ICIMTech50083.2020.9211239.
- [9] S. Chippagiri and P. Ravula, "Cloud-Native Development: Review of Best Practices and Frameworks for Scalable and Resilient Web Applications," vol. 8, pp. 13–21, 2021.
- [10] D. S. Lamba, S. Mishra, and S. Rastogi, "A Review Paper on Cloud Computing," *Int. J. Res. Publ. Rev.*, 2022, doi: 10.55248/gengpi.2022.3.3.7.
- [11] A. Katal, S. Dahiya, and T. Choudhury, "Energy efficiency in cloud computing data centers: a survey on software technologies," *Cluster Comput.*, 2023, doi: 10.1007/s10586-022-03713-0.
- [12] R. Vaño, I. Lacalle, P. Sowiński, R. S-Julian, and C. E. Palau, "Cloud-Native Workload Orchestration at the Edge: A Deployment Review and Future Directions," *Sensors*. 2023. doi: 10.3390/s23042215.
- [13] Vashudhar Sai Thokala, "Scalable Cloud Deployment and Automation for E-Commerce Platforms Using AWS, Heroku, and Ruby on Rails," *Int. J. Adv. Res. Sci. Commun. Technol.*, pp. 349–362, Oct. 2023, doi: 10.48175/IJARST-13555A.
- [14] M. Gopalsamy, S. Cyber, and S. Specialist, "Advanced Cybersecurity in Cloud Via Employing AI Techniques for Effective Intrusion Detection," *IJRAR*, vol. 8, no. 1, pp. 187–192, 2021.
- [15] R. Arora, S. Gera, and M. Saxena, "Impact of Cloud Computing Services and Application in Healthcare Sector and to provide improved quality patient care," *IEEE Int. Conf. Cloud Comput. Emerg. Mark. (CCEM)*, NJ, USA, 2021, pp. 45–47, 2021.
- [16] M. S. Rajeev Arora, Sheetal Gera, "Mitigating Security Risks on Privacy of Sensitive Data used in Cloud-based Mitigating Security Risks on Privacy of Sensitive Data used in Cloud-based ERP Applications," *8th Int. Conf. "Computing Sustain. Glob. Dev."*, pp. 458–463, 2021.
- [17] R. Bishukarma, "Scalable Zero-Trust Architectures for Enhancing Security in Multi-Cloud SaaS Platforms," *Int. J. Adv. Res. Sci. Commun. Technol.*, vol. 3, no. 3, pp. 1308–1319, 2023, doi: 10.48175/IJARST-14000S.
- [18] Ramesh Bishukarma, "Privacy-preserving based encryption techniques for securing data in cloud computing environments," *Int. J. Sci. Res. Arch.*, vol. 9, no. 2, pp. 1014–1025, Aug. 2023, doi: 10.30574/ijrsra.2023.9.2.0441.
- [19] A. and P. Khare, "Cloud Security Challenges : Implementing Best Practices for Secure SaaS Application Development," *Int. J. Curr. Eng. Technol.*, vol. 11, no. 6, pp. 669–676, 2021, doi: https://doi.org/10.14741/ijcet/v.11.6.11.
- [20] S. Murri, "Data Security Environments Challenges and Solutions in Big Data," *Int. J. Curr. Eng. Technol.*, vol. 12, no. 6, pp. 565–574, 2022.
- [21] M. Szalay, P. Mátray, and L. Toka, "State Management for Cloud-Native Applications," *Electronics*, vol. 10, no. 4, 2021, doi: 10.3390/electronics10040423.
- [22] Q. Jiao, B. Xu, and Y. Fan, "Design of Cloud Native Application Architecture Based on Kubernetes," in *Proceedings - 2021 IEEE International Conference on Dependable, Autonomic and Secure Computing, International Conference on Pervasive Intelligence and Computing, International Conference on Cloud and Big Data Computing and International Conference on Cybe*, 2021. doi: 10.1109/DASC-PICom-CBDCCom-CyberSciTech52372.2021.00088.
- [23] P. Mendki, "Securing Cloud Native Applications Using Blockchain," in *2021 12th International Conference on Information and Communication Systems, ICICS 2021*, 2021. doi: 10.1109/ICICS52457.2021.9464583.
- [24] R. Chowdhury, C. Talhi, H. Ould-Slimane, and A. Mourad, "A Framework for Automated Monitoring and Orchestration of Cloud-Native applications," in *2020 International Symposium on Networks, Computers and Communications (ISNCC)*, IEEE, Oct. 2020, pp. 1–6. doi: 10.1109/ISNCC49221.2020.9297238.
- [25] T. Pradeep Pai and C. R. S. Kumar, "Building cloud native

application - Analysis for multi-component application deployment,” in *2021 International Conference on Computer*

Communication and Informatics, ICCCI 2021, 2021. doi: 10.1109/ICCCI50826.2021.9402422.